

What Is Claimed Is:

1 1. A method for facilitating typesafe software design while
2 supporting structured composition of a software system, comprising:
3 receiving a first invocation of the software system;
4 assigning a first context to the first invocation;
5 examining the first invocation to locate components of the first invocation;
6 registering a unique factory to build each component, wherein these
7 factories are registered using the first context; and
8 when a component is needed, building the component using a factory
9 associated with the component, whereby building the component after each
10 component has a registered factory eliminates potential problems with
11 initialization circularity.

1 2. The method of claim 1, further comprising:
2 receiving a second invocation of the software system;
3 assigning a second context to the second invocation;
4 examining the second invocation to locate components of the second
5 invocation;
6 registering a unique factory to build each component, wherein these
7 factories are registered using the second context; and
8 when a component is needed, building the component using a factory
9 associated with the component, whereby building the component after each
10 component has a registered factory eliminates problems with initialization
11 circularity.

1 3. The method of claim 2, wherein components from the second
2 invocation are not available to the first invocation.

1 4. The method of claim 1, further comprising providing an additional
2 factory for an extended component of the first invocation.

1 5. The method of claim 1, wherein registering the unique factory to
2 build each component involves placing a key and a related factory identifier into a
3 storage structure.

1 6. The method of claim 5, wherein building the component using the
2 factory associated with the component involves using the key to retrieve the
3 related factory identifier from the storage structure.

1 7. The method of claim 6, wherein the storage structure includes a
2 hash table.

1 8. A computer-readable storage medium storing instructions that
2 when executed by a computer cause the computer to perform a method for
3 facilitating typesafe software design while supporting structured composition of a
4 software system, the method comprising:

5 receiving a first invocation of the software system;
6 assigning a first context to the first invocation;
7 examining the first invocation to locate components of the first invocation;
8 registering a unique factory to build each component, wherein these
9 factories are registered using the first context; and

10 when a component is needed, building the component using a factory
11 associated with the component, whereby building the component after each
12 component has a registered factory eliminates potential problems with
13 initialization circularity.

1 9. The computer-readable storage medium of claim 8, the method
2 further comprising:
3 receiving a second invocation of the software system;
4 assigning a second context to the second invocation;
5 examining the second invocation to locate components of the second
6 invocation;
7 registering a unique factory to build each component, wherein these
8 factories are registered using the second context; and
9 when a component is needed, building the component using a factory
10 associated with the component, whereby building the component after each
11 component has a registered factory eliminates problems with initialization
12 circularity.

1 10. The computer-readable storage medium of claim 9, wherein
2 components from the second invocation are not available to the first invocation.

1 11. The computer-readable storage medium of claim 8, the method
2 further comprising providing an additional factory for an extended component of
3 the first invocation.

1 12. The computer-readable storage medium of claim 8, wherein
2 registering the unique factory to build each component involves placing a key and
3 a related factory identifier into a storage structure.

1 13. The computer-readable storage medium of claim 12, wherein
2 building the component using the factory associated with the component involves
3 using the key to retrieve the related factory identifier from the storage structure.

1 14. The computer-readable storage medium of claim 13, wherein the
2 storage structure includes a hash table.

1 15. An apparatus for facilitating typesafe software design while
2 supporting structured composition of a software system, comprising:
3 a receiving mechanism configured to receive a first invocation of the
4 software system;
5 an assigning mechanism configured to assign a first context to the first
6 invocation;
7 an examining mechanism configured to examine the first invocation to
8 locate components of the first invocation;
9 a registering mechanism configured to register a unique factory to build
10 each component, wherein these factories are registered using the first context; and
11 a building mechanism configured to build the component using a factory
12 associated with the component when a component is needed, whereby building
13 the component after each component has a registered factory eliminates potential
14 problems with initialization circularity.

1 16. The apparatus of claim 15, wherein:

2 the receiving mechanism is further configured to receive a second
3 invocation of the software system;
4 the assigning mechanism is further configured to assign a second context
5 to the second invocation;
6 the examining mechanism is further configured to examine the second
7 invocation to locate components of the second invocation;
8 the registering mechanism is further configured to register a unique factory
9 to build each component, wherein these factories are registered using the second
10 context; and
11 the building mechanism is further configured to build the component using
12 a factory associated with the component when a component is needed, whereby
13 building the component after each component has a registered factory eliminates
14 problems with initialization circularity.

1 17. The apparatus of claim 16, wherein components from the second
2 invocation are not available to the first invocation.

1 18. The apparatus of claim 15, further comprising a providing
2 mechanism configured to provide an additional factory for an extended
3 component of the first invocation.

1 19. The apparatus of claim 15, wherein registering the unique factory
2 to build each component involves placing a key and a related factory identifier
3 into a storage structure.

1 20. The apparatus of claim 19, wherein building the component using
2 the factory associated with the component involves using the key to retrieve the
3 related factory identifier from the storage structure.

1 21. The apparatus of claim 20, wherein the storage structure includes a
2 hash table.